

EXHIBIT F

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

U.S. Patent No. 7,292,844 – Nordstrom, Inc.

Claims 1, 25, 32, 37 and 46

Push Data LLC (“Push Data”) provides evidence of infringement of Claims 1, 25, 32, 37 and 46 of U.S. Patent No. 7,292,844 (hereinafter “the ‘844 patent”) by Nordstrom, Inc. (“Defendant” or “Nordstrom”). In support thereof, Push Data provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least the Nordstrom application as developed for mobile electronic devices. These claim charts demonstrate Nordstrom’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities, as Nordstrom has not yet provided any non-public information. An analysis of Nordstrom’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Push Data reserves the right to supplement this infringement analysis once such information is made available to Push Data. Furthermore, Push Data reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Push Data contends that Nordstrom directly infringes the ‘844 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement.

Unless otherwise noted, Push Data believes and contends that each element of each claim asserted herein is literally met through Nordstrom’s provision of the Infringing Instrumentalities. However, to the extent that Nordstrom attempts to allege that any asserted claim element is not literally met, Push Data believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Push Data did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Push Data asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Push Data reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Nordstrom. Push Data also reserves the right to amend this infringement analysis by citing other claims of the '844 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Push Data further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

Google Play

Games

Apps

Movies & TV

Books

Kids

Nordstrom

Nordstrom, Inc.

4.9★

14K reviews

1M Downloads

Install

This app is available on the Google Play Store

Create Wish Lists
to save and share items you're interested in.

N

Nordstrom

About this app

Requires Android

8.0 and up

Downloads

1,000,000+ downloads

Content rating

Everyone [Learn more](#)

Permissions

[View details](#)

Released on

May 11, 2015

Offered by

Nordstrom, Inc.

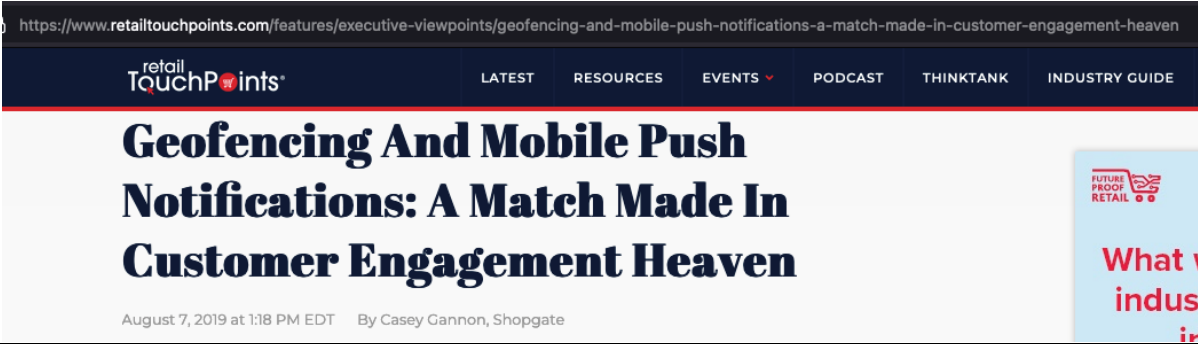
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

Claim #1	Accused Instrumentalities
1. A method for use in a mobile data network environment comprising a packet switched data network, one or more network servers, a plurality of mobile units including a particular mobile unit operated by a user, and a plurality of wireless packet access stations coupled to the packet switched data network, wherein each wireless packet access station provides wireless access services,	<p>A method is specified for use in a mobile data network environment that uses mobile internet data services such as provided by 3G, 4G, 5G and WiFi networks.</p> <p>The network servers can be any server that provides data services such as Internet services, and mobile Internet based services.</p> <p>The particular mobile unit can correspond to a smartphone or tablet operated by a user.</p> <p>The wireless packet access stations can be data nodes of 3G or 4G or 5G and/or WiFi access points.</p>
wherein the particular mobile unit comprises a processor, a memory, a graphical user interface, and at least one wireless air interface comprising a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions using a wireless packet data network protocol,	<p>The particular mobile unit can be any smartphone or tablet, and as such, will include a processor, a memory, and a graphical user interface. Additionally it will include a WiFi air interface subsystem that will include a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions using a wireless packet data network protocol, such as wireless internet protocols.</p> <p>The smartphone or tablet will also typically include a second wireless interface to support 3G or 4G or 5G data services and their related wireless data interactions.</p>
the particular mobile unit is configured to execute a plurality of application programs and to wirelessly receive an incoming communication from a particular one of the network servers of the	<p>The smartphone or tablet will run various application programs to include the Nordstrom application program after it has been downloaded.</p> <p>The Nordstrom application receives wireless push notification packets from a network server that originates at the Nordstrom server using an operating system push notification service. This corresponds to the incoming communication. For an overview of push notifications, see</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

<p>one or more network servers, read an application-program identifying field contained within the incoming communication to identify a particular application program resident on the particular mobile unit to which at least a portion of the incoming communication is to be directed, and</p>	<p>https://www.urbanairship.com/push-notifications-explained. Endnotes #3-#6 also provide further details about the Android-specific push notification services.</p> <p>Endnote #1 provides details on how each push notification coming into the Nordstrom application includes an app-specific device token [2]. The app-specific device token will be indicative of the Nordstrom application when push notifications are sent to the Nordstrom application.</p> <p>Endnote #3 describes the main types of push notification messages. The push notification message can be optionally displayed to the user. For example, a popup message can be displayed in the notifications tray or from a user interface supplied by the Nordstrom application.</p>
<p>the method comprising:</p>	<p>Note that all of the limitations above are the preamble which is describing the environment in which the method actions listed below are to be practiced.</p>
<p>causing the incoming communication to be wirelessly transmitted to the particular mobile unit, wherein the incoming communication includes the application program identifying field that identifies the particular application program and contains an address indicating from where further content is available to be downloaded,</p>	<p>The send-push notification type function calls made from the Nordstrom application server causes a push notification message to be wirelessly transmitted to the smartphone or tablet operated by a specified user.</p> <p>For details about the application-program identifying field, see the last sub-element of the preamble above and also see Endnote #1.</p> <p>Endnote #5 describes the data payload in push notification messages. Endnote #6 describes the format of push data messages. The push notification message can include a Uri object as its data payload. The Uri object serves an address that references external data not contained in the push notification message itself. A push data message can indicate an external address of external data not contained in the push data message itself. Otherwise, the push data message can include an indication to perform a data sync operation of the client to a server mailbox or any similar user account data structure. In these cases, the address corresponds to</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

	<p>an indication of a stream address that can be explicit or implicitly identified in the data notification’s payload data structure.</p> <p>The Nordstrom application supports push notifications. When the push notifications are enabled, push notifications are received. Push notifications cause further content related to the push notification to be downloaded or otherwise synchronized between the Nordstrom client-side App and the Nordstrom server-side application program that caused the push notification to be sent to the client-side App.</p>  A screenshot of a web article from Retail TouchPoints. The URL at the top is https://www.retailtouchpoints.com/features/executive-viewpoints/geofencing-and-mobile-push-notifications-a-match-made-in-customer-engagement-heaven. The page has a dark blue header with the Retail TouchPoints logo and navigation links: LATEST, RESOURCES, EVENTS (with a dropdown arrow), PODCAST, THINKTANK, and INDUSTRY GUIDE. The main headline is 'Geofencing And Mobile Push Notifications: A Match Made In Customer Engagement Heaven' in large, bold, black font. Below the headline, it says 'August 7, 2019 at 1:18 PM EDT' and 'By Casey Gannon, Shopgate'. On the right side, there is a vertical blue sidebar with the text 'What's in the future of retail?' and a small icon of a shopping cart with a plus sign.
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<p>https://www.retailtouchpoints.com/features/executive-viewpoints/geofencing-and-mobile-push-notifications-a-match-made-in-customer-engagement-heaven</p> <p>retail TouchPoints LATEST RESOURCES EVENTS PODCAST THINKTANK INDUSTRY GUIDE</p> <p>Geofencing Done Right In The Real World</p> <p>According to a survey from <i>Localitytics</i>, 42% of smartphone users said they'd use a retailer's app more if it sent them push notifications triggered by their present location. Geofencing technology is crucial to driving this type of engagement, giving retailers the ability to send contextually relevant push notifications on smartphones to help drive consumers to stores. Retailers can use geofencing-powered communication to optimize their mobile capabilities, targeting consumers where they are and staying at the top of their radar based on current location and location history.</p>  <p>Mizuno USA, a leading sports and athletic goods retailer, recently updated its mobile app to drive in-store traffic to its dealers. When a customer is near a dealer's store, Mizuno can send a geofence-powered push notification directly to a user's smartphone. Once the user clicks on it, he or she will be taken directly to Google Maps to easily locate the stores that are within a specified geofence radius. The same can be done as soon as a user is near a competitor's store, steering them away with a better promotion or offer. Retailers can even create a push notification campaign targeting users at a relevant sporting event, sending participants an incentive to visit a brick-and-mortar store while creating a positive association of the brand.</p> <p>Other innovative retailers and brands have also recently utilized geofencing to their advantage:</p> <ul style="list-style-type: none"> • McDonald's recently connected geofenced billboards to its in-app advertising on Waze to achieve 6.4 million mobile impressions and prompt consumers to visit a nearby location during their drive. • A Volvo Dealership in the New York Tri-State metro has utilized geofencing initiatives to target consumers in luxury markets and competitor dealerships, as well as consumers within their own dealership, to achieve a 140% increase in foot traffic. • Nordstrom utilizes geofencing to identify when loyal customers are within the store to offer hyper-personalized customer service. <p>What w indust in</p> <p>EXPLORE THE RI WORLD OF TOM</p>
the incoming communication is not a server response message	As discussed in the section directly above, the incoming communication that the Nordstrom server causes to be wirelessly transmitted is a push notification message.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

<p>sent in response to a client request message coupled from the particular mobile unit substantially just prior to the incoming communication being transmitted,</p>	<p>A push notification message is not a server response message sent in response to a client request message coupled from the particular mobile unit substantially just prior to the incoming communication being transmitted. For example, a push notification message is not a server response message sent in response to a client Get message in the HTML and related markup language protocols.</p>
<p>the portion of the incoming communication is coupled to the particular application program at least partially via a virtual communication session implemented at one or more layers below an application layer,</p>	<p>In Android based systems and devices, push notification messages are sent using an operating system push notification service. All push notification messages are sent over Android/FCM/GCM Push Sessions that use Transport Layer Security connections. This TLS security for push sessions is mandated by the Android operating system.</p> <p>See Endnote#2 for a discussion of the virtual session aspects of TLS. Note that TLS is located at a layer below the application layer and above the transport layer, often called the sockets layer.</p> <p>A virtual communication session is a session that has a full handshake sequence that is used to establish connection parameters, and an abbreviated handshake sequence that is used to resume the virtual session connection from an inactive or dormant state to an active state whereby new payload data can be sent via the virtual session once again.</p> <p>TLS connections are virtual sessions because they establish parameters in an initial handshake procedure to determine session parameters such as device tokens, and then reuse these virtual session parameters later to wake up and reuse the TLS session in an abbreviated handshake procedure.</p>
<p>the virtual communication session is configured to be transitioned from an initial active state to an inactive state, and later to be transitioned from the inactive</p>	<p>In Android/FCM/GCM wireless push notification services, push notification message is sent via a virtual communication session (TLS session) that is configured to be transitioned from an initial active state to an inactive state, and later to be transitioned from the inactive state back to the active state, and when the virtual communication session is in the active state, the push</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

state back to the active state, and when the virtual communication session is in an instance of the active state, the portion of the incoming communication is coupled to the particular application program via the virtual communication session;	<p>notification message can be coupled to the mobile unit via the virtual communication session (TLS session, see especially Endnote #2).</p> <p>As per Endnote #1, the wireless push session is established and used over and over again over the life of the application. As per Endnote #2, TLS is used to reactivate the wireless push session each time a new push needs to be sent from the push server to the client-side app. This reactivation and push message sending is caused by the Nordstrom application server making an API call to cause the push notification message to be sent.</p>
receiving a client-request packet wirelessly coupled from the particular mobile unit, the client-request packet indicating a request to download the further content and including the address; and	<p>The server-side application program receives the client-request packet. This client-request packet indicates a request to download the further content. The request packet indicates information contained in a Uri object or else an indication to pull from a stream for syncing the client-side app to the server-side application.</p> <p>See Endnotes #5 and #6 for a discussion of the Uri object and requests to sync the client-side app to the server-side application. Downloading of the further content is performed in response thereto.</p>
sending the further content to the particular mobile unit in response to the client-request packet;	<p>The Nordstrom server provides the Nordstrom application service to the Nordstrom application running on the user's smartphone or tablet. The Nordstrom server will respond to the client-request packet by sending the further content to the Nordstrom application.</p> <p>See the causing element above for more information concerning the specific push message and further content sent by Nordstrom server-side application.</p>
wherein the incoming communication acts as a notification to allow the particular mobile unit to download the further content by transmitting the client-request packet and	<p>The incoming communication is a push message that allows the user to thereby selectively download the further content. Upon receipt of the client-request packet, the server-side application causes the further content located at the address identified in the incoming communication to be sent to the client-side app in the mobile unit.</p>

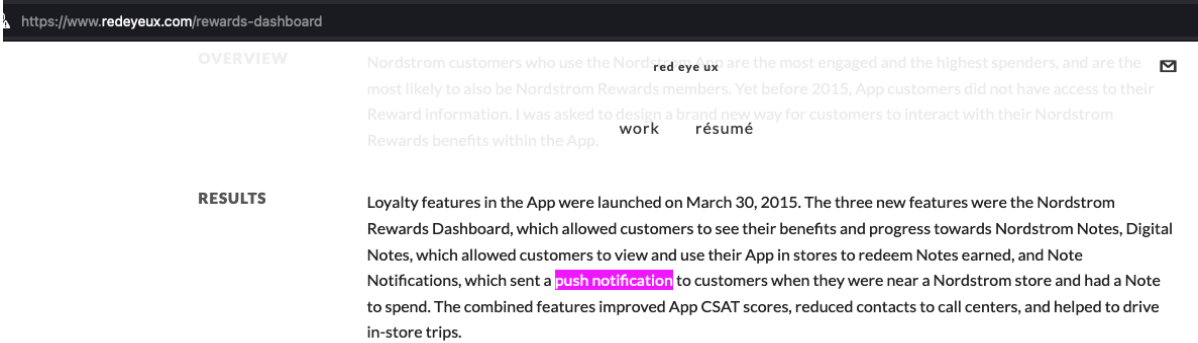
CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

receiving the further content located at the address contained in the incoming communication.	See the causing element above for to see how the Nordstrom system sends a notification that contains information indicative of an address from where the further content can be downloaded into or otherwise sourced from the server side-application to be sent to the client-side app.
---	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

Claim #25	Accused Instrumentalities
<p>A method for use in a system comprising a wireless handheld device that includes first and second transceivers configured to be selectively wirelessly coupled to a first wireless packet network access station provided by a base station of a cellular communications network and a second wireless packet network access station that uses a low-power wireless local area network air interface whose coverage area is substantially smaller than a coverage area provided by the first wireless packet network access station,</p>	<p>A method is specified for use in a mobile data network environment that uses mobile internet data services such as provided by 3G, 4G, 5G and WiFi networks.</p> <p>The wireless handheld device can be any smartphone or tablet, and as such, will include a processor, a memory, and a graphical user interface.</p> <p>The smartphone or tablet will also typically include a first transceiver to support 3G or 4G or 5G data services and their related wireless data interactions.</p> <p>Additionally smartphone or tablet will include a second transceiver corresponding to a WiFi transceiver subsystem that will include a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions using a wireless packet data network protocol, such as wireless internet protocols.</p>
<p>wherein the first and second wireless packet network access stations are coupled via a packet switched data network to a remote server system, and the wireless handheld device is configured to communicate with the remote server system via a selected one of the first and second wireless packet network access stations and via the packet switched data network,</p>	<p>The remote server system includes a server such as the Nordstrom server that provides the Nordstrom service to a specified user's smartphone or tablet type device.</p> <p>The first wireless packet network access station can be data nodes of 3G or 4G or 5G cellular data networks. The second wireless packet network access station will be a WiFi access point.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

the method comprising:	Note that the limitations above are the preamble which is describing the environment in which the method actions listed below are to be practiced.
the remote server system receiving a first request coupled thereto from the wireless handheld device via the first wireless packet network access station;	<p>The remote server system includes the server that provides the Nordstrom application service. An application program that provides the client side of the Nordstrom application service provides a first request for data updates. This first request is sent to the server when the user configures the phone to receive notifications and to allow background mode operations.</p> <p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for notifications. This allows notifications to be allowed for the app and this menu can be used to determine whether the user will see when the notifications come in or not.</p>  <p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for background mode data. From the App setting menu, Usage category, then Mobile Data, there is a toggle for “Allow Background mode usage.”</p> <p>Also, from the app settings menu, Usage category, and then “Battery” there is a toggle for “Allow background activity.” Also in the App settings menu, App settings category, there is a toggle to allow Notifications.</p> <p>https://ting.com/blog/ting-tip-for-android-control-which-apps-use-background-mobile-data/</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<p><i>What is background data?</i> Basically, background data means that an app is using data even when you're not actively using the app. Sometimes called background syncing, background data can keep your apps updated with the latest notifications like status updates, Snapchat stories and Tweets.</p> <p>Does background app refresh use data? Apps refresh in the background to regularly check for notifications. This means, when you get an email, message or Tweet, it's delivered right to your device, whether you're on Wi-Fi or mobile data.</p> <p>When you turn off background data and prevent apps from apps refreshing in the background, you won't get any notifications. You'll have to open the app to check and see if you have any notifications.</p>
the remote server system transmitting a server response to the wireless handheld device, the server response including an indication of availability of content related to the first request;	<p>As discussed in directly above, the Nordstrom app in the android phone can be configured to receive notifications and enable background mode processing in response to the remote notifications.</p> <p>Endnote #3 describes the main message types used for remote notifications in Android. For example, the FCM Notification Messages go to the notifications tray when the Nordstrom app is in the background mode FCM Data Messages cause the onMessageReceived() method in the client side app to be awakened to process in the incoming notification when it is of the form of a FCM Data Message.</p> <p>Endnote #6 provides the details of processing FCM Data messages and similar Android or GCM data messages when they are of a form similar to FCM Data Messages.</p> <p>As discussed in Endnote #6, each FCM Data Message includes a data payload that is effectively a JSON element (Java Script Object Notation)(https://en.wikipedia.org/wiki/JSON). The contents of the data payload indicate to the client side app an indication of availability of content</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

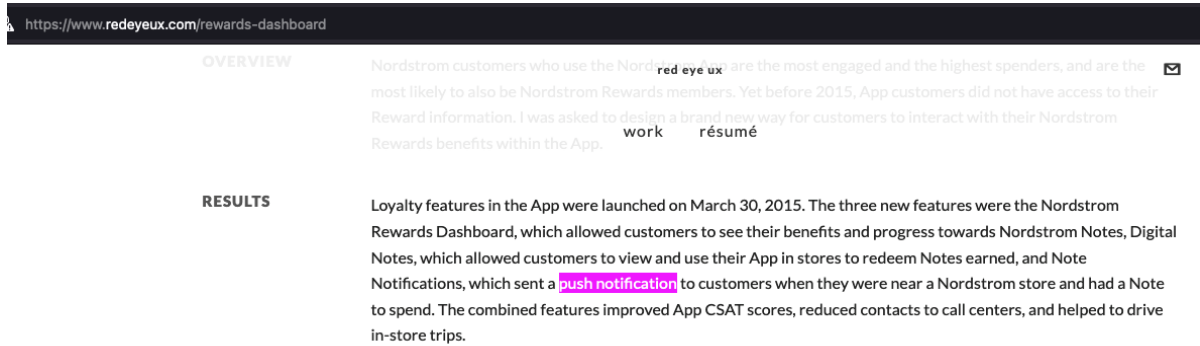
	<p>related to the first request. For example, when notifications and background mode processing are enabled, the incoming FCM Data Message formatted notification will tell the client-side app that the server has new information such so that a client-server sync operation is needed, or that other related information, such as parameters related to an incoming VoIP call are available.</p> <p>Server sync operations are preformed, for example when the server mailbox, message box, or user-account box has changed and is out of date relative to the version displayable by the client-side app.</p> <p>Background mode used with FCM Data Messages is common in messaging apps, email apps, chat room apps, chat apps, geofencing apps, gaming apps, VoIP apps, Waze, Maps, etc.</p>
the remote server system receiving a second request coupled thereto from the wireless handheld device, wherein the second request is automatically generated by the wireless handheld device in response to the server response, without requiring user action, and the second request is coupled via the second wireless packet network access station; and	<p>See Endnote #6. When the incoming notification that is formatted as an FCM Data Message is received, the onMessageReceived() method of Nordstrom's client-side app is activated. When the onMessageReceived() method is activated, the mobile unit will automatically send a second request to the server to request the additional content as discussed in the section directly above to be sent to the Nordstrom client side app.</p> <p style="text-align: center;"><i>Note from a damages perspective, many of the times, the second request of this claim element will be received at the remote server via the WiFi air interface.</i></p>
the remote server coupling the available content related to the first request to the wireless handheld device, via the second wireless packet access station.	<p>See Endnote #6. When the notification in the form of an FCM Data message is received, the client-side app wakes up by activating its onMessageReceived() method. This allows the client-side app to automatically request a server sync operation or a similar operation to pull down additional content from the Nordstrom server. When the Nordstrom server receives this request, it couples the available content that is sendable by the notifications and background mode processing capabilities of the Nordstrom system to the client-side app.</p> <p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for notifications and for background mode data. Go to the App setting</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<p>menu, Usage category, then click on Mobile Data, then see the toggle for “Allow Background mode usage.” Then from the app settings menu, Usage category, also go to “Battery” then see the toggle for “Allow background activity.” From the app settings menu, App settings category, click on Notifications. This allows notifications to be allowed for the app and this menu can be used to determine whether the user will see when the notifications come in or not.</p> <p>Note from a damages perspective, many of the times, the second request of this claim element will be received at the remote server via the WiFi air interface.</p>
--	--

Claim #32	Accused Instrumentalities
A method for use in a mobile data network environment comprising a packet switched data network, one or more servers including a particular server, a plurality of mobile units including a particular mobile unit operated by a user, at least one wireless packet access station that provides wireless access services and is coupled to the packet switched data network,	<p>A method is specified for use in a mobile data network environment that uses mobile internet data services such as provided by 3G, 4G, 5G and WiFi networks.</p> <p>The particular server is a server such as the Nordstrom server that provides the Nordstrom service to a specified user’s smartphone or tablet type device.</p> <p>The particular mobile unit can be a smartphone or tablet operated by any specified user.</p> <p>The wireless packet access stations can be data nodes of 3G or 4G or 5G and/or WiFi access points.</p>
wherein the particular mobile unit comprises a processor, a memory, and at least one wireless air interface comprising a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions,	<p>The particular mobile unit can be any smartphone or tablet, and as such, will include a processor, a memory, and a graphical user interface. Additionally it will include a WiFi air interface subsystem that will include a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions using a wireless packet data network protocol, such as wireless internet protocols.</p> <p>The smartphone or tablet will also typically include a second wireless interface to support 3G or 4G or 5G data services and their related wireless data interactions.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

the method comprising:	Note that the limitations above are the preamble which is describing the environment in which the method actions listed below are to be practiced.
the particular server receiving a first request coupled thereto from the particular mobile unit via the particular wireless packet network access station and via the packet switched data network;	<p>The particular server system includes the server that provides the Nordstrom application service. An application program that provides the client side of the Nordstrom application service provides a first request for data updates. This first request is sent to the server when the user configures the phone to receive notifications and to allow background mode operations.</p>  <p>The screenshot shows a web browser address bar with the URL https://www.redeyeux.com/rewards-dashboard. The page content includes a section titled "OVERVIEW" with text about Nordstrom customers using the Nordstrom app, and a section titled "RESULTS" with text about loyalty features launched on March 30, 2015. The text in the "RESULTS" section mentions "push notification" in a pink box.</p> <p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for notifications and for background mode data. Go to the App setting menu, Usage category, then click on Mobile Data, then see the toggle for “Allow Background mode usage.” Then from the app settings menu, Usage category, also go to “Battery” then see the toggle for “Allow background activity.” From the app settings menu, App settings category, click on Notifications. This allows notifications to be allowed for the app and this menu can be used to determine whether the user will see when the notifications come in or not.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<p>https://ting.com/blog/ting-tip-for-android-control-which-apps-use-background-mobile-data/</p> <p><i>What is background data?</i> Basically, background data means that an app is using data even when you're not actively using the app. Sometimes called background syncing, background data can keep your apps updated with the latest notifications like status updates, Snapchat stories and Tweets.</p> <p>Does background app refresh use data? Apps refresh in the background to regularly check for notifications. This means, when you get an email, message or Tweet, it's delivered right to your device, whether you're on Wi-Fi or mobile data.</p> <p>When you turn off background data and prevent apps from apps refreshing in the background, you won't get any notifications. You'll have to open the app to check and see if you have any notifications.</p>
<p>the particular server coupling a server response to the particular mobile unit, the server response including an indication of availability of content related to the first request;</p>	<p>As discussed in the section directly above, the Nordstrom app in the Android phone can be configured to receive notifications and enable background mode processing in response to the remote notifications.</p> <p>Endnote #3 describes the main message types used for remote notifications in Android. For example, the FCM Notification Messages go to the notifications tray when the Nordstrom app is in the background mode FCM Data Messages cause the onMessageReceived() method in the client side app to be awakened to process in the incoming notification when it is of the form of a FCM Data Message.</p> <p>Endnote #6 provides the details of processing FCM Data messages and similar Android or GCM data messages when they are of a form similar to FCM Data Messages.</p> <p>As discussed in Endnote #6, each FCM Data Message includes a data payload that is effectively a JSON element (Java Script Object Notation)(https://en.wikipedia.org/wiki/JSON). The</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<p>contents of the data payload indicate to the client side app an indication of availability of content related to the first request. For example, when notifications and background mode processing are enabled, the incoming FCM Data Message formatted notification will tell the client-side app that the server has new information such so that a client-server sync operation is needed, or that other related information, such as parameters related to an incoming VoIP call are available.</p> <p>Server sync operations are preformed, for example when the server mailbox, message box, or user-account box has changed and is out of date relative to the version displayable by the client-side app.</p> <p>Background mode used with FCM Data Messages is common in messaging apps, email apps, chat room apps, chat apps, geofencing apps, gaming apps, VoIP apps, Waze, Maps, etc.</p>
the particular server receiving a second request coupled thereto from the particular mobile unit, wherein the second request is automatically generated by the particular mobile unit in response to the server response, without requiring user action; and	<p>See Endnote #6. When the incoming notification that is formatted as an FCM Data Message is received, the onMessageReceived() method of Nordstrom's client-side app is activated. When the onMessageReceived() method is activated, the mobile unit will automatically send a second request to the server to request the additional content as discussed in the section directly above to be sent to the Nordstrom client side app.</p> <p>The onMessageReceived() method is activated without requiring user action. Alert information can be displayed to the user, but no user action is required.</p>
the particular server coupling the available content related to the first request to the particular mobile unit.	<p>See Endnote #6. When the notification in the form of an FCM Data message is received, the client-side app wakes up by activating its onMessageReceived() method. This allows the client-side app to automatically request a server sync operation or a similar operation to pull down additional content from the Nordstrom server. When the Nordstrom server receives this request, it couples the available content that is sendable by the notifications and background mode processing capabilities of the Nordstrom system to the client-side app.</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

	<p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for notifications and for background mode data. Go to the App setting menu, Usage category, then click on Mobile Data, then see the toggle for “Allow Background mode usage.” Then from the app settings menu, Usage category, also go to “Battery” then see the toggle for “Allow background activity.” From the app settings menu, App settings category, click on Notifications. This allows notifications to be allowed for the app and this menu can be used to determine whether the user will see when the notifications come in or not.</p>
--	--

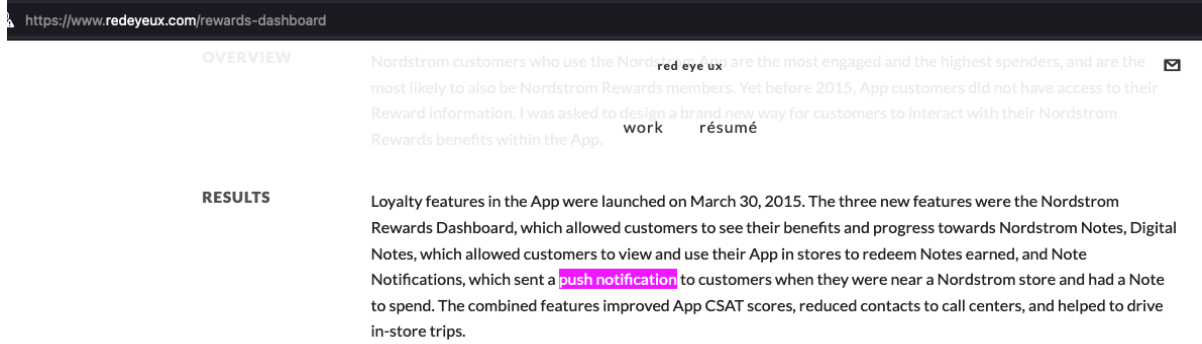
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

Claim #37	Accused Instrumentalities
A method for use in a mobile data network environment comprising a packet switched data network,	A method is specified for use in a mobile data network environment that uses mobile internet data services such as provided by 3G, 4G, 5G and WiFi networks.
one or more servers including a virtual session server and a particular network server,	<p>A virtual session server is a hardware and/or software server module that establishes and maintains a virtual session. A virtual session is a session that has a full handshake sequence that is used to establish connection parameters, and an abbreviated handshake sequence that is used to resume the virtual session connection from an inactive or dormant state to an active state whereby new payload data can be sent via the virtual session once again.</p> <p>See Endnote#2 for a discussion of the virtual session aspects of TLS.</p> <p>The particular server is a server such as the Nordstrom server that provides the Nordstrom service to a specified user's smartphone or tablet type device.</p>
a mobile unit operated by a user, one or more wireless packet access stations including a particular wireless packet access station that provides wireless access services and is coupled to the packet switched data network, wherein the mobile unit comprises a processor, a memory, and at least one wireless air interface comprising a wireless transmitter, a wireless receiver, and a protocol stack adapted to process wireless packet data transactions,	<p>The particular mobile unit can be a smartphone or tablet operated by any specified user.</p> <p>The wireless packet access stations can be data nodes of 3G or 4G or 5G and/or WiFi access points.</p> <p>The particular mobile unit can be any smartphone or tablet, and as such, will include a processor, a memory, and a graphical user interface. Additionally it will include a WiFi air interface subsystem that will include a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions using a wireless packet data network protocol, such as wireless internet protocols.</p> <p>The smartphone or tablet will also typically include a second wireless interface to support 3G or 4G or 5G data services and their related wireless data interactions.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

the method comprising:	Note that the limitations above are the preamble which is describing the environment in which the method actions listed below are to be practiced.
establishing a communication session between the mobile unit and the virtual session server via the particular wireless packet access station, using the at least one wireless air interface;	<p>Mobile applications in Android systems communicate with their application server via TLS connections. These TLS connections are established at the time the app is installed or launched and can be resumed at a later time using a session token.</p> <p>See Endnote#2 for a discussion of TLS.</p> <p>The Nordstrom application and the Nordstrom server communicate over a TLS session in order to provide secure client-server communications between the client app and the server.</p>
the particular network server receiving a first request coupled thereto from the mobile unit at least partially via the communication session;	<p>The particular network server system includes the server that provides the Nordstrom application service. An application program that provides the client side of the Nordstrom application service provides a first request for data updates. This first request is sent to the server when the user configures the phone to receive notifications and to allow background mode operations.</p> <p>The first request is sent via the client-server connection established between the Nordstrom client-side app and the Nordstrom server-side application. As discussed in the section directly above, in Android systems, the established client-server communication session is a TLS session.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<div data-bbox="663 250 1858 592">  <p>The screenshot shows a dashboard titled "https://www.redeyeux.com/rewards-dashboard". It has two main sections: "OVERVIEW" and "RESULTS".</p> <p>OVERVIEW: Nordstrom customers who use the Nordstrom app are the most engaged and the highest spenders, and are the most likely to also be Nordstrom Rewards members. Yet before 2015, App customers did not have access to their Reward information. I was asked to design a brand new way for customers to interact with their Nordstrom Rewards benefits within the App.</p> <p>RESULTS: Loyalty features in the App were launched on March 30, 2015. The three new features were the Nordstrom Rewards Dashboard, which allowed customers to see their benefits and progress towards Nordstrom Notes, Digital Notes, which allowed customers to view and use their App in stores to redeem Notes earned, and Note Notifications, which sent a push notification to customers when they were near a Nordstrom store and had a Note to spend. The combined features improved App CSAT scores, reduced contacts to call centers, and helped to drive in-store trips.</p> </div> <p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for notifications and for background mode data. Go to the App setting menu, Usage category, then click on Mobile Data, then see the toggle for “Allow Background mode usage.” Then from the app settings menu, Usage category, also go to “Battery” then see the toggle for “Allow background activity.” From the app settings menu, App settings category, click on Notifications. This allows notifications to be allowed for the app and this menu can be used to determine whether the user will see when the notifications come in or not.</p> <p>https://ting.com/blog/ting-tip-for-android-control-which-apps-use-background-mobile-data/</p> <p><i>What is background data?</i></p> <p>Basically, background data means that an app is using data even when you’re not actively using the app. Sometimes called background syncing, background data can keep your apps updated with the latest notifications like status updates, Snapchat stories and Tweets.</p> <p>Does background app refresh use data?</p> <p>Apps refresh in the background to regularly check for notifications. This means, when you get an email, message or Tweet, it’s delivered right to your device, whether you’re on Wi-Fi or mobile data.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<p>When you turn off background data and prevent apps from apps refreshing in the background, you won't get any notifications. You'll have to open the app to check and see if you have any notifications.</p>
<p>the particular network server coupling a server response to the mobile unit, the server response including an indication of availability of content related to the first request;</p>	<p>As discussed in the section directly above, the Nordstrom app in the Android phone can be configured to receive notifications and enable background mode processing in response to the remote notifications.</p> <p>Endnote #3 describes the main message types used for remote notifications in Android. For example, the FCM Notification Messages go to the notifications tray when the Nordstrom app is in the background mode FCM Data Messages cause the onMessageReceived() method in the client side app to be awakened to process in the incoming notification when it is of the form of a FCM Data Message.</p> <p>Endnote #6 provides the details of processing FCM Data messages and similar Android or GCM data messages when they are of a form similar to FCM Data Messages.</p> <p>As discussed in Endnote #6, each FCM Data Message includes a data payload that is effectively a JSON element (Java Script Object Notation)(https://en.wikipedia.org/wiki/JSON). The contents of the data payload indicate to the client side app an indication of availability of content related to the first request. For example, when notifications and background mode processing are enabled, the incoming FCM Data Message formatted notification will tell the client-side app that the server has new information such so that a client-server sync operation is needed, or that other related information, such as parameters related to an incoming VoIP call are available.</p> <p>Server sync operations are preformed, for example when the server mailbox, message box, or user-account box has changed and is out of date relative to the version displayable by the client-side app.</p> <p>Background mode used with FCM Data Messages is common in messaging apps, email apps, chat room apps, chat apps, geofencing apps, gaming apps, VoIP apps, Waze, Maps, etc.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

the particular network server receiving a second request coupled thereto from the mobile unit, wherein the second request is automatically generated by the mobile unit in response to the server response, without requiring user action; and	<p>See Endnote #6. When the incoming notification that is formatted as an FCM Data Message is received, the onMessageReceived() method of Nordstrom's client-side app is activated. When the onMessageReceived() method is activated, the mobile unit will automatically send a second request to the server to request the additional content as discussed in the section directly above to be sent to the Nordstrom client side app.</p> <p>The onMessageReceived() method is activated without requiring user action. Alert information can be displayed to the user, but no user action is required.</p>
the particular network server coupling the available content related to the first request to the mobile unit;	<p>See Endnote #6. When the notification in the form of an FCM Data message is received, the client-side app wakes up by activating its onMessageReceived() method. This allows the client-side app to automatically request a server sync operation or a similar operation to pull down additional content from the Nordstrom server. When the Nordstrom server receives this request, it couples the available content that is sendable by the notifications and background mode processing capabilities of the Nordstrom system to the client-side app.</p> <p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for notifications and for background mode data. Go to the App setting menu, Usage category, then click on Mobile Data, then see the toggle for "Allow Background mode usage." Then from the app settings menu, Usage category, also go to "Battery" then see the toggle for "Allow background activity." From the app settings menu, App settings category, click on Notifications. This allows notifications to be allowed for the app and this menu can be used to determine whether the user will see when the notifications come in or not.</p>
wherein the communication session is configured to be transitioned between an active state and an inactive state; and	See endnote #2. The TLS session has a Session ID that is used in a Client Hello message in an abbreviated handshake to resume the TLS session without the need for a full handshake sequence to be repeated to determine a new session ID.
wherein the first request is received by the virtual session server and forwarded to the	See the first method step section above in this claim chart and also Endnote#2. A secure Client-Server TLS session is established between the Nordstrom application and the Nordstrom server. Subsequent client server transactions are sent over the previously

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

particular network server during a first active state of the communication session, and the second request is received by the particular network server during a second active state of the communication session which follows an inactive state of the communication session.	<p>established TLS session using the Session ID in a Client Hello packet to resume the TLS session back into an active state after being in an inactive state.</p> <p>The TLS session is in the inactive state when the session is dormant and resumes the active state when the client begins the background download in response to the remote notification sent in the FCM Data Message format with a data payload that indicates to perform a client-server or similar download operation.</p>
---	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

Claim #46	Accused Instrumentalities
<p>A method for use in a mobile data network environment comprising a packet switched data network, one or more servers including a particular server, a plurality of mobile units including a particular mobile unit operated by a user, at least one wireless packet access station that provides wireless access services and is coupled to the packet switched data network,</p>	<p>A client-side method is specified for use in a mobile data network environment that uses mobile internet data services such as provided by 3G, 4G, 5G and WiFi networks.</p> <p>The particular server is a server such as the Nordstrom server that provides the Nordstrom service to a specified user's smartphone or tablet type device.</p> <p>The particular mobile unit can be a smartphone or tablet operated by any specified user.</p> <p>The wireless packet access stations can be data nodes of 3G or 4G or 5G and/or WiFi access points.</p>
<p>wherein the particular mobile unit comprises a processor, a memory, and at least one wireless air interface comprising a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions,</p>	<p>The particular mobile unit can be any smartphone or tablet, and as such, will include a processor, a memory, and a graphical user interface. Additionally it will include a WiFi air interface subsystem that will include a wireless transmitter, a wireless receiver, and a protocol stack adapted to process packet data transactions using a wireless packet data network protocol, such as wireless internet protocols.</p> <p>The smartphone or tablet will also typically include a second wireless interface to support 3G or 4G or 5G data services and their related wireless data interactions.</p>
<p>the method comprising:</p>	<p>Note that the limitations above are the preamble which is describing the environment in which the method actions listed below are to be practiced.</p>
<p>receiving from the user of the particular mobile unit an indication of a type of information of interest to the user;</p>	<p>In the Nordstrom application service, the user identifies specific types of information of interest where the user would like to receive notifications.</p>

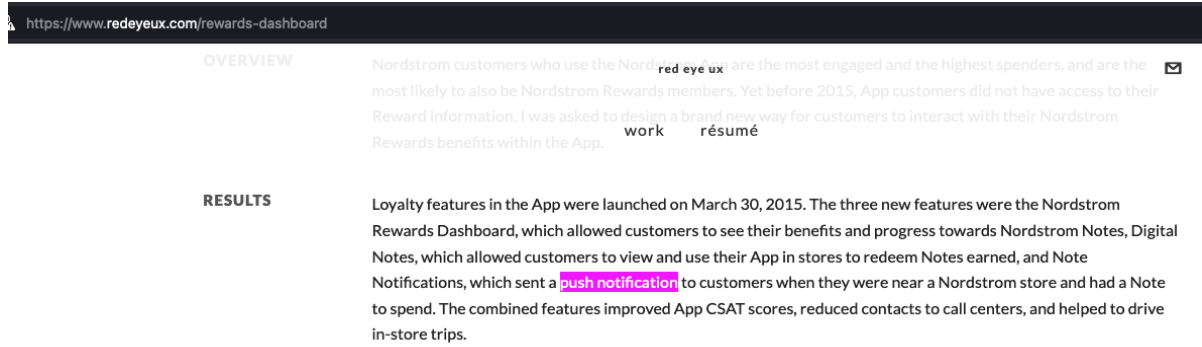
CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

	<p>This may be implicit based on user actions with the app. For example, in a messaging app, once the user identifies a distant client endpoint to be a contact, messages from that client will be known to be of interest to that user.</p> <p>For example, the application program collects information from the user's use of the application program that Nordstrom uses as an indication of the type of information that is of interest to the user, as shown below:</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	 <p>The screenshot shows a webpage from retailTouchpoints.com. The article title is "Geofencing Done Right In The Real World". The text discusses a survey from Localytics stating that 42% of smartphone users would use a retailer's app more if it sent push notifications based on their location. It highlights the importance of geofencing for driving engagement and sending relevant push notifications. An e-book titled "2,400 Shoppers Told Us What They Want. Are You Listening?" is featured with a "DOWNLOAD" button. The article also mentions Mizuno USA's updated mobile app for driving in-store traffic and lists examples of other retailers using geofencing, including McDonald's, Volvo, and Nordstrom.</p>
the particular mobile unit receiving from the particular	The particular network server system includes the server that provides the Nordstrom application service. An application program that provides the client side of the Nordstrom application

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

<p>server a server response that includes an indication of availability of information corresponding to the type of information of interest to the user;</p>	<p>service provides a first request for data updates. This first request is sent to the server when the user configures the phone to receive notifications and to allow background mode operations.</p> <p>The first request is sent by the particular mobile unit via the client-server connection established between the Nordstrom client-side app and the Nordstrom server-side application.</p>  <p>In an Android smartphone or tablet, the Apps settings menu of the Nordstrom app allows the app to be configured for notifications and for background mode data. Go to the App setting menu, Usage category, then click on Mobile Data, then see the toggle for “Allow Background mode usage.” Then from the app settings menu, Usage category, also go to “Battery” then see the toggle for “Allow background activity.” From the app settings menu, App settings category, click on Notifications. This allows notifications to be allowed for the app and this menu can be used to determine whether the user will see when the notifications come in or not.</p> <p>https://ting.com/blog/ting-tip-for-android-control-which-apps-use-background-mobile-data/</p> <p><i>What is background data?</i></p> <p>Basically, background data means that an app is using data even when you’re not actively using the app. Sometimes called background syncing, background data can keep your apps updated with the latest notifications like status updates, Snapchat stories and Tweets.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	<p>Does background app refresh use data?</p> <p>Apps refresh in the background to regularly check for notifications. This means, when you get an email, message or Tweet, it's delivered right to your device, whether you're on Wi-Fi or mobile data.</p> <p>When you turn off background data and prevent apps from apps refreshing in the background, you won't get any notifications. You'll have to open the app to check and see if you have any notifications.</p>
<p>and the particular mobile unit sending to the particuiar (sp) server a request, wherein the second request is automatically generated by the particular mobile unit, without requiring user action; and</p> <p>Note: Certificate of correction to strike the typo word "second" in "second request" – clerical typo. The second request is in a lot of other claims where there are first and second requests.</p>	<p>Endnote #3 describes the main message types used for remote notifications in Android. For example, the FCM Notification Messages go to the notifications tray when the Nordstrom app is in the background mode FCM Data Messages cause the onMessageReceived() method in the client side app to be awakened to process in the incoming notification when it is of the form of a FCM Data Message.</p> <p>Endnote #6 provides the details of processing FCM Data messages and similar Android or GCM data messages when they are of a form similar to FCM Data Messages.</p> <p>As discussed in Endnote #6, each FCM Data Message includes a data payload that is effectively a JSON element (Java Script Object Notation) (https://en.wikipedia.org/wiki/JSON). The contents of the data payload indicate to the client side app an indication of availability of content related to the first request. For example, when notifications and background mode processing are enabled, the incoming FCM Data Message formatted notification will tell the client-side app that the server has new information such so that a client-server sync operation is needed, or that other related information, such as parameters related to an incoming VoIP call are available.</p> <p>Server sync operations are preformed, for example when the server mailbox, message box, or user-account box has changed and is out of date relative to the version displayable by the client-side app.</p> <p>Background mode used with FCM Data Messages is common in messaging apps, email apps, chat room apps, chat apps, geofencing apps, gaming apps, VoIP apps, Waze, Maps, etc.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

	See Endnote #6. When the incoming notification that is formatted as an FCM Data Message is received, the onMessageReceived() method of Nordstrom's client-side app is activated. When the onMessageReceived() method is activated, the mobile unit will automatically send a second request to the server to request the additional content as discussed in the section directly above to be sent to the Nordstrom client side app.
the particular mobile unit receiving from the particular server the information corresponding to the type of information of interest to the user.	As discussed in Endnote #6 , when the FCM Data message is received by the client-side app, the client side onMessageReceived() method of the application program wakes up (becomes activated). The onMessageReceived() method or related code in the client-side app can then automatically receive (download) the content related to the user interest in the background.

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

Endnote#1 - app-specific device token

<https://help.pushwoosh.com/hc/en-us/articles/360000364923-What-is-a-Device-token->

What is a Device token?

Question:

What is a Device token?

Answer:

Push token (device token) - is a **unique key for the app-device combination** which is issued by the Apple or **Google push notification** gateways. It allows gateways and push notification providers to route messages and ensure the notification is delivered only to the **unique app-device combination** for which it is intended.

iOS device push tokens are strings with 64 hexadecimal symbols. Push token example:

03df25c845d460bcdad7802d2vf6fc1dfde97283bf75cc993eb6dca835ea2e2f

Make sure that iOS push tokens you use when targeting specific devices in your API requests are in **lower case**.

Android device push tokens can differ in length (usually below 255 characters), and usually start with **APA...** Push token example:

APA91bFoi3lMMre9G3XzR1LrF4ZT82_15MsMdEICogXSLB8-
MrdkRuRQFwNI5u8Dh0cI90ABD3BOKnxkEla8cGdisbDHI5cVlkZah5QUhSAxzx4Roa7b4xy9tvx9iNSYw-
eXBYYd8k1XKf8Q_Qq1X9-x-U-Y79vdPq

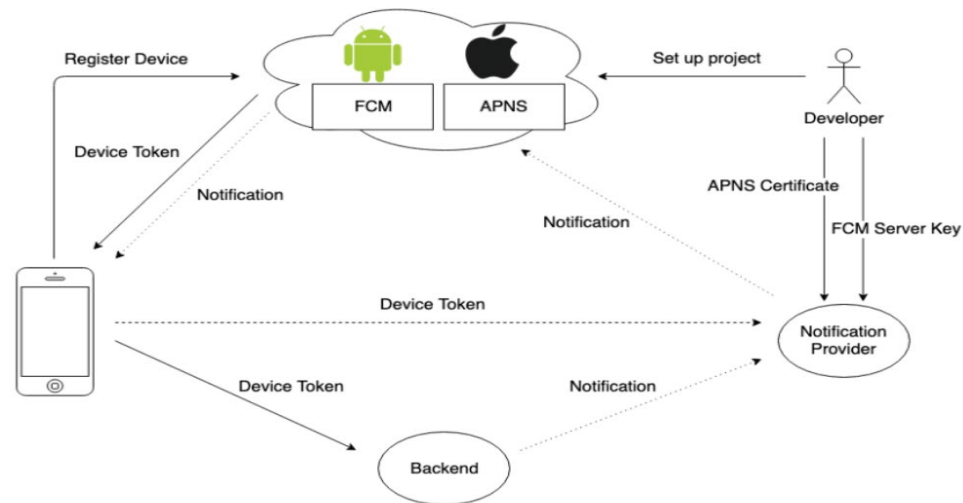
.....

Note: The **Android device push tokens** correspond to the **app-specific device token** terminology used in the claim charts.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

<https://dev.to/jakubkoci/react-native-push-notifications-313i>

Architecture



CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

In the above Architecture, the “backend” corresponds to the backend of the Application Server. The Device token corresponds to a specific App running on a specific device. That is what is meant by the **app-specific device token** in the claim charts.

<https://firebase.google.com/docs/cloud-messaging/android/first-message>

“Access the registration token

To send a message to a specific device, you need to know that device's registration token. Because you'll need to enter the token in a field in the Notifications console to complete this tutorial, make sure to copy the token or securely store it after you retrieve it.

On initial startup of your app, the **FCM SDK** generates a **registration token for the client app** instance. If you want to target single devices or create device groups, you'll need to access this token by extending `FirebaseMessagingService` and overriding `onNewToken`.

This section describes how to retrieve the token and how to monitor changes to the token. Because the token could be rotated after initial startup, you are strongly recommended to retrieve the latest updated registration token.

The registration token may change when:

- The app deletes Instance ID
- The app is restored on a new device
- The user uninstalls/reinstall the app
- The user clears app data.”

<https://firebase.google.com/docs/cloud-messaging/concept-options>

For example, here is a JSON-formatted notification message in an IM app. The user can expect to see a message with the title "Portugal vs. Denmark" and the text "great match!" on the device:

```
{
  "message": {
    "token": "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...", ←-- App-specific token
```

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

```
"notification": {
  "title": "Portugal vs. Denmark",
  "body": "great match!"
}
}
```

<https://firebase.google.com/docs/cloud-messaging/android/client>

Retrieve the current registration token

When you need to retrieve the current token, call `FirebaseInstanceId.getInstance().getInstanceId()`:

```
FirebaseInstanceId.getInstance().getInstanceId()
    .addOnCompleteListener(new OnCompleteListener<InstanceIdResult>() {
        @Override
        public void onComplete(@NonNull Task<InstanceIdResult> task) {
            if (!task.isSuccessful()) {
                Log.w(TAG, "getInstanceId failed", task.getException());
                return;
            }

            // Get new Instance ID token
            String token = task.getResult().getToken();

            // Log and toast
            String msg = getString(R.string.msg_token_fmt, token);
            Log.d(TAG, msg);
            Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();
        }
    });
```

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

[MainActivity.java](#)

Monitor token generation

The `onNewToken` callback fires whenever a new token is generated.

```
/**
 * Called if InstanceID token is updated. This may occur if the security of
 * the previous token had been compromised. Note that this is called when the InstanceID token
 * is initially generated so this is where you would retrieve the token.
 */
@Override
public void onNewToken(String token) {
    Log.d(TAG, "Refreshed token: " + token);

    // If you want to send messages to this application instance or
    // manage this apps subscriptions on the server side, send the
    // Instance ID token to your app server.
    sendRegistrationToServer(token);
}
```

After you've obtained the token, you can send it to your app server and store it using your preferred method. See the [Instance ID API reference](#) for full detail on the API.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

Endnote#2 - Transport Layer Security and Virtual Sessions

<https://developer.android.com/training/articles/security-ssl>

“The Secure Sockets Layer (SSL)—now technically known as [Transport Layer Security \(TLS\)](#)—is a **common building block for encrypted communications between clients and servers.**”

<https://android-developers.googleblog.com/2018/04/protecting-users-with-tls-by-default-in.html>

“**Android** is committed to keeping users, their devices, and their data safe. One of the ways that we keep data safe is **by protecting all data that enters or leaves an Android device with Transport Layer Security (TLS) in transit.**”

http://abbas.rpanah.ir/publications/conext2017_tls_paper.pdf - However, other protocols such as secure email (42 apps) and **Google’s Cloud Messaging service for push notifications** (9 apps) [11, 47] also **use TLS**. A History of TLS Support in Android: Android has supported TLS 1.0 since its first version released in 2008 and TLS 1.1 and TLS 1.2 since 2012.

<https://developer.ibm.com/customer-engagement/docs/watson-marketing/ibm-engage-2/tls-1-2-migration-for-mobile-push-clients/>

What will happen on devices that are unable to support TLS 1.2?

Devices which do not support TLS 1.2 will be unable to connect to our WCA servers. This will prevent users of those devices from:

- Registering new mobile user IDs
- **Updating push tokens**
- Receiving inbox messages
- **Receiving In-app messages**

As the above link shows, the creation of the App IDs of Endnote #1 are linked to the TLS protocol being run on the TLS-enabled Push-Notification channel.

<https://tools.ietf.org/html/rfc5246> - TLS 1.2 - Has fast session resumption, section F.1.4

<https://tools.ietf.org/html/rfc5077> - This version introduces server tickets, like Android Device Push Tokens

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

Abstract

This document describes a mechanism that enables the Transport **Layer Security (TLS) server to resume sessions** and avoid keeping per-client session state. **The TLS server encapsulates the session state into a ticket and forwards it to the client.** The client can subsequently **resume a session** using the obtained ticket.

3. Protocol

This specification describes a mechanism to distribute encrypted session-state information in the form of a ticket. The ticket is created by a TLS server and sent to a TLS client. The TLS client presents the ticket to the TLS server to **resume a session**.

Endnote#3 – Android/FCM/GCM Notification Message Types

<https://firebase.google.com/docs/cloud-messaging/concept-options>

About FCM messages

Firebase Cloud Messaging (FCM) offers a broad range of messaging options and capabilities. The information in this page is intended to help you understand the different types of FCM messages and what you can do with them.

Message types

With FCM, you can send two types of messages to clients:

- **Notification messages**, sometimes thought of as "display messages." These are handled by the FCM SDK automatically.
- **Data messages**, which are handled by the client app.

Notification messages contain a predefined set of user-visible keys. Data messages, by contrast, contain only your user-defined custom key-value pairs. Notification messages can contain an optional data payload. Maximum payload for both message types is 4KB, except when sending messages from the Firebase console, which enforces a 1024 character limit.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

Use scenario	How to send
<p>Notification message</p> <p>FCM automatically displays the message to end-user devices on behalf of the client app. Notification messages have a predefined set of user-visible keys and an optional data payload of custom key-value pairs.</p>	<ol style="list-style-type: none"> 1. In a trusted environment such as Cloud Functions or your app server, use the Admin SDK or the FCM Server Protocols: Set the notification key. May have optional data payload. Always collapsible. 2. Use the Notifications composer: Enter the Message Text, Title, etc., and send. Add optional data payload by providing Custom data.
<p>Data message</p> <p>Client app is responsible for processing data messages. Data messages have only custom key-value pairs with no reserved key names (see below).</p>	<p>In a trusted environment such as Cloud Functions or your app server, use the Admin SDK or the FCM Server Protocols: Set the data key only.</p>

Use notification messages when you want FCM to handle displaying a notification on your client app's behalf. Use data messages when you want to process the messages on your client app.

FCM can send a notification message including an optional data payload. In such cases, FCM handles displaying the notification payload, and the client app handles the data payload. (...)

Notification messages are delivered to the notification tray when the app is in the background. For apps in the foreground, messages are handled by a callback function. (...)

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

App behavior when receiving messages that include both notification and data payloads depends on whether the app is in the background or the foreground (...).

- **When in the background**, apps receive the notification payload in the notification tray, and only handle the data payload when the user taps on the notification. [Data messages go straight to the client app and wake it up. That is, both the data message and its optional data payload are delivered directly to the client app with no need for the user to tap on the user interface, although sounds may be triggered and badges and short messages may be flash-displayed to alert the user.]
- **When in the foreground**, your app receives a message object with both payloads available

<https://firebase.google.com/docs/cloud-messaging/android/receive>

Handling messages

(...) [onMessageReceived() = executed when a notification message or a data message is received.]

- **Notification messages delivered when your app is in the background.** In this case, the notification is delivered to the device's system tray. A user tap on a notification opens the app launcher by default.
- **Messages with both notification and data payload, when received in the background.** In this case, the notification is delivered to the device's system tray, and the data payload is delivered in the extras of the intent of your launcher Activity. In summary:

App state	Notification	Data	Both
Foreground	onMessageReceived	onMessageReceived	onMessageReceived

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

Background	System tray	onMessageReceived	Notification: system tray Data: in extras of the intent .
------------	-------------	-------------------	---

Endnote#4 – Actions Taken Upon Notification-Message User Tap

<https://developer.android.com/guide/topics/ui/notifiers/notifications>

Notifications Overview

A notification is a message that Android displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification. (...)

Notification actions

Although it's not required, every notification should open an appropriate app activity when tapped. In addition to this default notification action, you can add action buttons that complete an app-related task from the notification (often without opening an activity), as shown in figure 9.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

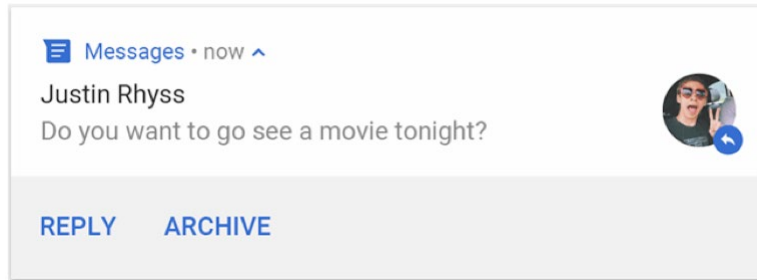


Figure 9. A notification with action buttons

Starting in Android 10 (API level 29), the platform can automatically generate action buttons with suggested **intent**-based actions.

[Note: When the user taps the notification or an action button on the notification, an **Intent** can be executed as an activity, a service, or a receive-broadcast operation. The **Intent**, delivered as a **Pending Intent** identifies the program code to be executed upon **user tap** of the **Notification**. The **Intent** and **Pending Intent** executed upon **user-tap** the **Notification** is described below.]

<https://developer.android.com/guide/components/intents-filters>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

Intents and Intent Filters

An [Intent](#) is a messaging object you can use to request an action from another [app component](#). Although intents facilitate communication between components in several ways, there are three fundamental use cases:

- **Starting an activity**

An [Activity](#) represents a single screen in an app. You can start a new instance of an [Activity](#) by passing an [Intent](#) to [startActivity\(\)](#). The [Intent](#) describes the activity to start and carries any necessary data. (...)

- **Starting a service**

A [Service](#) is a component that performs operations in the background without a user interface. (...)

(...) You can start a service to perform a one-time operation (such as downloading a file) by passing an [Intent](#) to [startService\(\)](#). The [Intent](#) describes the service to start and carries any necessary data.

- **Delivering a broadcast**

A broadcast is a message that any app can receive. The system delivers various broadcasts for system events, such as when the system boots up or the device starts charging. (...)

Using a pending intent

A [PendingIntent](#) object is a wrapper around an [Intent](#) object. The primary purpose of a [PendingIntent](#) is to grant permission to a foreign application to use the contained [Intent](#) as if it were executed from your app's own process.

Major use cases for a **pending intent** include the following:

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

- Declaring an intent to be executed when the user performs an action with your [Notification](#) (the Android system's [NotificationManager](#) executes the [Intent](#)). (...)

[Y]ou must declare the intended component type when you create the [PendingIntent](#) by calling the respective creator method:

- [PendingIntent.getActivity\(\)](#) for an [Intent](#) that starts an [Activity](#).
- [PendingIntent.getService\(\)](#) for an [Intent](#) that starts a [Service](#).
- [PendingIntent.getBroadcast\(\)](#) for an [Intent](#) that starts a [BroadcastReceiver](#).

For more information about using pending intents, see the documentation for each of the respective use cases, such as in the [Notifications](#) and [App Widgets](#) API guides.

[The Notification Message can carry a PendingIntent object, such that upon user tap, an activity or service is launched. The activity generally corresponds to a UI screen in the client app, while the service corresponds to a background process in the app, as used to download a file or the like.]

<https://developer.android.com/training/notify-user/build-notification>

Set the notification's tap action

Every notification should respond to a tap, usually to open an activity in your app that corresponds to the notification. To do so, you must specify a content intent defined with a [PendingIntent](#) object and pass it to [setContentIntent\(\)](#). (...)

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

To add an action button, pass a [PendingIntent](#) to the [addAction\(\)](#) method. This is just like setting up the notification's default tap action, except instead of launching an activity, you can do a variety of other things such as start a [BroadcastReceiver](#) that performs a job in the background so the action does not interrupt the app that's already open.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

Endnote#5 – URI Data Field in Notification and Data Messages

<https://developer.android.com/guide/components/intents-filters>

The primary information contained in an **Intent** is the following:

Component name - The name of the component to start. (...)

Action - A string that specifies the generic action to perform (such as *view* or *pick*). (...)

Data -The URI (a **Uri** object) that references the data to be acted on and/or the MIME type of that data. The type of data supplied is generally dictated by the intent's action. For example, if the action is **ACTION_EDIT**, the data should contain the **URI** of the document to edit. (...)

Category - A string containing additional information about the kind of component that should handle the intent. (...)

Extras - Key-value pairs that carry additional information required to accomplish the requested action. Just as some actions use particular kinds of data URIs, some actions also use particular extras. (...)

[Note, the **Intent** object, passed in the **PendingIntent** object in the Notification Message includes a **Data** field which is a **Uri** object. (URI = Uniform Resource Identifier.) The Uri object unambiguously identifies items such as local resources in the client, external files stored on the application server, or external data sources in an external content provider database. The **Uri** object can resolve to a MIME type, and can also be resolved to find external information like web pages from content providers and the like. The links below explain how build the **Uri** object and how the **Uri** object can resolved.]

<https://developer.android.com/training/app-links/deep-linking>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

Create Deep Links to App Content

When a clicked link or programmatic request invokes a web URI intent, the Android system tries each of the following actions, in sequential order, until the request succeeds:

1. Open the user's preferred app that can handle the URI, if one is designated.
2. Open the only available app that can handle the URI.
3. Allow the user to select an app from a dialog.

Follow the steps below to create and test links to your content. You can also use the [App Links Assistant](#) in Android Studio to add Android App Links.

To create a link to your app content, add an intent filter that contains these elements and attribute values in your manifest:

[<action>](#) (...)

[<data>](#) Add one or more [<data>](#) tags, each of which represents a URI format that resolves to the activity. At minimum, the [<data>](#) tag must include the [android:scheme](#) attribute.

You can add more attributes to further refine the type of URI that the activity accepts. For example, you might have multiple activities that accept similar URIs, but which differ simply based on the path name. In this case, use the [android:path](#) attribute or its `pathPattern` or `pathPrefix` variants to differentiate which activity the system should open for different URI paths. (...)

`<intent-filter>`

...

`<data android:scheme="https" android:host="www.example.com" />`

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
 U.S. Patent No. 7,292,844

```
<data android:scheme="app" android:host="open.my.app" />
</intent-filter>
```

It might seem as though this supports only `https://www.example.com` and `app://open.my.app`. However, it actually supports those two, plus these: `app://www.example.com` and `https://open.my.app`.

Once you've added intent filters with URIs for activity content to your app manifest, Android is able to route any [Intent](#) that has matching URIs to your app at runtime.

<https://developer.android.com/reference/android/content/ContentResolver>

[The **Uri** object can be evaluated to see what the Uri is referencing. Some example classes and public methods are given below.]

Nested classes

```
class acquireContentProviderClient\(Uri uri\)
```

Returns a [ContentProviderClient](#) that is associated with the [ContentProvider](#) that services the content at **uri**, starting the provider if necessary.

```
call\(Uri uri, String method, String arg, Bundle extras\)
```

Call a provider-defined method.

```
requestSync\(Account account, String authority, Bundle extras\)
```

Start an asynchronous sync operation

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

<https://firebase.google.com/docs/cloud-messaging/concept-options>

Here is an example of a normal priority message sent via the FCM HTTP v1 protocol to notify a magazine subscriber that new content is available to download:

```
{
  "message": {
    "topic": "subscriber-updates",
    "notification": {
      "body": "This week's edition is now available.",
      "title": "NewsMagazine.com",
    },
    "data": {
      "volume": "3.21.15",
      "contents": "http://www.news-magazine.com/world-week/21659772"
    },
    "android": {
      "priority": "normal"
    },
    "apns": {
      "headers": {
        "apns-priority": "5"
      }
    },
    "webpush": {
      "headers": {
        "Urgency": "high"
      }
    }
  }
}
```

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

Endnote#6 – Background Refresh and other Activities in response to Data Messages

<https://support.kochava.com/sdk-integration/android-sdk-integration/android-push-notification/>

onMessageReceived:

Kochava Push uses the **FCM Data Message** format. This format ensures all push messages are received by the “onMessageReceived” method no matter if the app is in the foreground or background. The fields included in the RemoteMessage data are as follows. A Kochava push will always include the “kochava” key, other fields may vary.

1. **kochava** – Campaign tracking information to be added to the notification bundle and sent with the Push Open event.
2. **silent** – If this key is present the push was used for tracking uninstalls and should be ignored.
3. **title** – Notification title.
4. **message** – Notification message body.
5. **icon_resource_id** – Mapping string to an app internal drawable resource.
6. **link** – Launch deeplink Uri.

[The link above explains that a **FCM Data Message** will cause onMessageReceived() to fire/execute in an app when the app is in the background mode. See Endnote #1 for a discussion of the FCM Data Message. That is, when an app is in background mode, a FCM Notification Message will go to the notifications tray, but the FCM Data Message will cause the app to awaken by activating

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

the `onMessageReceived()` method. Depending on how `onMessageReceived()` is configured, this can cause background application refresh to occur, like the client syncing a mailbox or a chat session with a server, or could cause an application like a VoIP telecommunications app to be launched.]

<https://www.semicolonworld.com/question/44020/how-to-handle-notification-when-app-in-background-in-firebase>

How to handle notification when app in background in Firebase

When the app is in background and notification arrives then the default notification comes and doesn't run my code of `onMessageReceived`.

Here is my `onMessageReceived` code. This invokes if my app is running on foreground, not when app in background. How to run this code when the app is in background too?

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

- June 2018 Answer -

You have to make sure there is not a "notification" keyword anywhere in the message. Only include "data", and the app will be able to handle the message in `onMessageReceived`, even if in background or killed.

Using Cloud Functions:

```
const message = {  
  token: token_id, // obtain device token id by querying data in firebase  
  data: {  
    title: "my_custom_title",  
    body: "my_custom_body_message"  
  }  
}  
  
return admin.messaging().send(message).then(response => {  
  // handle response  
});
```

Then in your `onMessageReceived()`, in your class extending `com.google.firebase.messaging.FirebaseMessagingService` :

```
if (data != null) {  
  Log.d(TAG, "data title is: " + data.get("title");  
  Log.d(TAG, "data body is: " + data.get("body");  
}  
  
// build notification using the body, title, and whatever else you want.
```

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

[The link below answers the same question as the above link. The bottom line is to use a data message and to use OnMessageReceived() to get the content out of the data message and to take actions based on the extracted data.]

<https://stackoverflow.com/questions/37711082/how-to-handle-notification-when-app-in-background-in-firebase>

[The links below explain how data messages can be prioritized to wake up an app and schedule a worker process to perform background sync with a server side application process like an email box or a chat session or a gaming session, a messaging app, or a maps type app.

<https://firebase.google.com/docs/cloud-messaging/concept-options>

You have two options for assigning delivery priority to downstream messages on Android: normal and high priority. Delivery of normal and high priority messages works like this:

- **Normal priority.** This is the default priority for [data messages](#). Normal priority messages are delivered immediately when the app is in the foreground. When the device is in Doze, delivery may be delayed to conserve battery. **For less time-sensitive messages, such as notifications of new email, keeping your UI in sync, or syncing app data in the background,** choose normal delivery priority.

When receiving a normal priority message on Android that requests a **background data sync for your app**, you can schedule a task with [WorkManager](#) to handle it when the network is available.

- **High priority.** FCM attempts to deliver high priority messages immediately, allowing the FCM service to wake a sleeping device when necessary and to run some limited processing (including very limited network access). (...)

If you need to **sync** for additional in-app content on Android, you can schedule a task with [WorkManager](#) to handle that in the background.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

<https://developer.android.com/topic/libraries/architecture/workmanager>

Use WorkManager for Deferrable and Reliable Work

WorkManager is intended for work that is **deferrable**—that is, not required to run immediately—and required to **run reliably** even if the app exits or the device restarts. For example:

- Sending logs or analytics to backend services
- Periodically **syncing application data with a server**

[See also:] <https://developer.android.com/topic/libraries/architecture/workmanager/basics>

[High priority push messages can also be used, for example to alert the user to an incoming VoIP call.]

<https://documentation.onesignal.com/docs/voip-notifications>

Android VoIP Setup

Android does not have the concept of "VoIP Push" the same as iOS. Notifications will just work, including **data-only messages where you can start an Activity** instead of showing a push.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

If you need to call custom class for showing native UI for example, you would use the `Android NotificationExtenderService` so you can override the notification and show your custom UI or start your custom Activity.

[The activity started in response to the data message could be selected from the android call processing API as linked below.]

<https://developer.android.com/guide/topics/connectivity/telecom/selfManaged>

[Android background mode processing is described in the links below]

<https://ting.com/blog/ting-tip-for-android-control-which-apps-use-background-mobile-data/>

<https://swiftsenpai.com/testing/send-silent-push-notifications/>

<https://developer.android.com/training/notify-user/navigation?>

[Android CompactNotifications can be used to start any desired activity from a compact mode data message.]

<https://documentation.onesignal.com/docs/service-extensions#notification-extender-service>

Notification Extender Service

Android

Note! This requires writing native Android code

Set up the `NotificationExtenderService` if you want to do one of the following:

- Receive data in the background with or without displaying a notification.
- Override specific notification settings depending on client side app logic such as custom accent color, vibration pattern, or other any other `NotificationCompat` options available. See [Android's documentation on the NotificationCompat options.](#)



CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF NORDSTROM
U.S. Patent No. 7,292,844

OneSignal also supports sending additional data along with a notification as key value pairs. You can read this additional data when a notification

<https://developer.android.com/reference/androidx/core/app/NotificationCompat>